[U-Boot] [PATCH 1/5] CVE-2019-13103: disk: stop infinite recursion in DOS Partitions

Paul Emge paulemge at forallsecure.com
Mon Jul 8 23:37:03 UTC 2019

- Previous message: [U-Boot] [PATCH 0/2] efi_loader: Fix inconsistencies in efi_add_memory_map usage
- Next message: [U-Boot] [PATCH 2/5] CVE-2019-13105: ext4: fix double-free in ext4_cache_read
- **Messages sorted by:** [date] [thread] [subject] [author]

```
part_get_info_extended and print_partition_extended can recurse infinitely
while parsing a self-referential filesystem or one with a silly number of
extended partitions. This patch adds a limit to the number of recursive
partitions.

Signed-off-by: Paul Emge <paulemge at forallsecure.com>
---
 disk/part_dos.c | 18 ++++++++++++++++++
 1 file changed, 18 insertions(+)

diff --git a/disk/part_dos.c b/disk/part_dos.c
index 936cee0d36..aae9d95906 100644
--- a/disk/part_dos.c
+++ b/disk/part_dos.c
@@ -23,6 +23,10 @@

 #define DOS_PART_DEFAULT_SECTOR 512

+/* should this be configurable? It looks like it's not very common at all
+ * to use large numbers of partitions */
+#define MAX_EXT_PARTS 256
+
 /* Convert char[4] in little endian format to the host format integer
  */
 static inline unsigned int le32_to_int(unsigned char *le32)
@@ -126,6 +130,13 @@ static void print_partition_extended(struct blk_desc *dev_desc,
        dos_partition_t *pt;
        int i;

+       /* set a maximum recursion level */
+       if (part_num > MAX_EXT_PARTS)
+       {
+               printf("** Nested DOS partitions detected, stopping **\n");
+               return;
+    }
+
        if (blk_dread(dev_desc, ext_part_sector, 1, (ulong *)buffer) != 1) {
                printf ("** Can't read partition table on %d:" LBAFU " **\n",
                        dev_desc->devnum, ext_part_sector);
@@ -191,6 +202,13 @@ static int part_get_info_extended(struct blk_desc *dev_desc,
        int i;
        int dos_type;

+       /* set a maximum recursion level */
+       if (part_num > MAX_EXT_PARTS)
+       {
+               printf("** Nested DOS partitions detected, stopping **\n");
+               return -1;
+    }
```

```
+
        if (blk_dread(dev_desc, ext_part_sector, 1, (ulong *)buffer) != 1) {
                printf ("** Can't read partition table on %d:" LBAFU " **\n",
                        dev_desc->devnum, ext_part_sector);
--
2.20.1
```

- Previous message: [U-Boot] [PATCH 0/2] efi_loader: Fix inconsistencies in efi_add_memory_map usage
- Next message: [U-Boot] [PATCH 2/5] CVE-2019-13105: ext4: fix double-free in ext4_cache_read
- **Messages sorted by:** [date] [thread] [subject] [author]

More information about the U-Boot mailing list